**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

| | | |
|---|---|---|
| In re application of: **Casabona et al.** | § | Group Art Unit: **2193** |
| | § | |
| Serial No.: **09/696,399** | § | Examiner: **Vu, Tuan A.** |
| | § | |
| Filed: **October 25, 2000** | § | Attorney Docket No.: **DE9-1999-0075US1** |
| | § | |
| For: **Method and System for Maintaining Software Via Network** | | |

37945
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

<u>Certificate of Transmission Under 37 C.F.R. § 1.8(a)</u>
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (571) 273-8300 on August 22, 2005.

By: Jennifer Pilcher
Jennifer Pilcher

## TRANSMITTAL DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
ENCLOSED HEREWITH:

- Appeal Brief (37 C.F.R. 41.37).

A fee of $500.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0457. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0457. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0457.

Respectfully submitted,

Duke W. Yee
*Registration No. 34,285*
**YEE & ASSOCIATES, P.C.**
P.O. Box 802333
Dallas, Texas 75380
(972) 385-8777
ATTORNEY FOR APPLICANTS

Docket No. DE9-1999-0075US1 *PATENT*

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Casabona et al.** §
§ Group Art Unit: **2193**
Serial No. **09/696,399** §
§ Examiner: **Vu, Tuan A.**
Filed: **October 25, 2000** §
§
For: **Method and System for Maintaining Software Via Network** §
§

**Commissioner for Patents**
**P.O. Box 1450**
**Alexandria, VA 22313-1450**

**Certificate of Transmission Under 37 C.F.R. § 1.8(a)**
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (571) 273-8300 on August 22, 2005.

By:
Jennifer Pilcher

## APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on June 21, 2005.

The fees required under § 41.20(B)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

## REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation.

## RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

## STATUS OF CLAIMS

### A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-16

### B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: none
2. Claims withdrawn from consideration but not canceled: none
3. Claims pending: 1-16
4. Claims allowed: none
5. Claims rejected: 1-16
6. Claims objected to: none

### C. CLAIMS ON APPEAL

The claims on appeal are: 1-16

## STATUS OF AMENDMENTS

No amendment after final was filed for this case.

## SUMMARY OF CLAIMED SUBJECT MATTER

### A. CLAIM 1 - INDEPENDENT

Claim 1 is directed to a technique for providing updates to software programs in devices using a hierarchical approach, where a hierarchy of overlay repositories such as country-level and system-level are provided.

Specifically, Claim 1 recites a method for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution. The client computer systems are connectable with the at least one central software maintenance institution via a network. The method includes steps of (i) providing product information for a product in the network system in order to make the product information available for the plurality of client systems; and (ii) performing a software maintenance action for the client site product by downloading data required for the software maintenance action from a sequence of repositories, where the sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product. The first subset of files is specific for a given client system, and data downloaded from the top-level repository is different from data downloaded from the local-level repository. The data downloaded from both the top-level repository and the local-level repository is used by the given client system in performing the software maintenance action (Specification page 21, line 16 – page 24, line 10; Figures 5A-5B, all steps).

### B. CLAIM 4 - INDEPENDENT

Claim 4 is directed to a technique for providing updates to software programs in devices using a hierarchical approach, where a hierarchy of overlay repositories such as country-level and system-level are provided. An update request from a client device traverses the hierarchy to create a customized list of files to download.

Specifically, Claim 4 is directed to a method for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution. The client computer systems are connectable with the at least one central software maintenance institution via a network. The method includes steps of (a) providing product information for a product in the network system in order to make the product information available for the plurality of client systems; (b) performing a software maintenance action for the product from a client site by downloading data required for the software maintenance action from a sequence of repositories, where the sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, where the first subset of files is specific for a given client system. This 'performing a software maintenance' step includes sub-steps of (i) generating an input list of files downloadable from the sequence of repositories; (ii) generating a list of files present on the target client system; (iii) comparing the list of files downloadable from the sequence of repositories with the list of files present on the target client system; and (iv) downloading a plurality of files, where the plurality of files includes only files which are not yet present in the target client system (Specification page 21, line 16 – page 24, line 10; Figures 5A-5B, all steps).

### C. CLAIM 7 - INDEPENDENT

Claim 7 is a system claim of similar scope to the method claim recited in Claim 1, and the summary of Claim 1 given above is equally applicable to Claim 7, and thus is incorporated herewith to provide the summary of Claim 7.

### D. CLAIM 12 - INDEPENDENT

Claim 12 is a program product claim of similar scope to the method claim recited in Claim 1, and the summary of Claim 1 given above is equally applicable to Claim 12, and thus is incorporated herewith to provide the summary of Claim 12.

## GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

### A. GROUND OF REJECTION 1 (Claims 1-16)

Claims 1-16 stand rejected under 35 U.S.C. § 103 as being unpatentable over MacInnis, USPN 6,487,723, in view of Saether et al., USPN 6,405,219.

## ARGUMENT

### A. GROUND OF REJECTION 1 (Claims 1-16)

### A.1. Claims 1-3, 7-9, 12 and 13

With respect to Claim 1, Appellants urge that none of the cited references teach or suggest the claimed step of "downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system". In rejecting Claim 1, the Examiner states that Macinnis teaches:

> "performing a software maintenance action for the product (e.g. Fig. 3, col. 5, lines 11-25) from the client side by downloading the data required for said maintenance from a combination of a top-level repository storing a set of files for the product (e.g. system 200, Fig. 2) and a local-level repository storing a first subset of files for the product (e.g. terminal 203, 204, Fig. 2; internal table – col. 4, line 45 to col. 5, line 25 – Note: internal table and local storage of files or hardware modules at terminal reads on local-level repository storing subset of files specific to a given client), wherein the first subset of files is specific for a given client system" (emphasis added by Appellants)

Appellants show error in such assertion, as Claim 1 specifically recites a client site, and that data is downloaded *from* a top-level repository *and* a local-level repository. Thus, Claim 1 recites (1) a client site, (2) a top-level repository, and (3) a local-level repository. To the extent MacInnis terminal is interpreted to be the claimed 'local-level repository', then MacInnis does not teach the claimed 'client site'. Perhaps even more importantly, information in the MacInnis terminal, including information in the internal tables and local storage, is not downloaded. Claim 1 recites a top-level repository, a local-level repository and a client site, and as a part of performing a maintenance operation for a client site product, data is downloaded from *both* the top-level repository and the local-level repository. Contrary to the Examiner's assertion, MacInnis does not teach a client site, and that data is downloaded *from* (1) a top-level repository *and* (2) a local-level repository as a part of performing a maintenance operation for the client site product (the two repositories being different from the client site). Thus, the Examiner's reasoning in rejection Claim 1 is shown to be in error.

Appellants further show error in the Examiner's rejection of Claim 1. The Examiner acknowledges that the MacInnis reference does not teach downloading from a sequence of repositories (Appellant note: this appears to be directly contrary to the Examiner's position, quoted above), but cites Saether as teaching distribution of software in sequence from a more global server to more secondary global servers before updating the target machines. Appellants urge that Saether teaches the updating of source files on content servers (column 1, lines 10-14), and has nothing whatsoever to do with maintaining software products in client computer systems. Thus, even to the extent Saether may show multiple servers, where source content is copied from one server to another, such technique is not useful for managing or maintaining programs in client computer systems. By analogy, a teaching of how to distribute power from a power plant to a local neighborhood substation is very different from how to manage wiring or distribute power internal to a household. The system characteristics, equipment, power levels, etc. are very different between such diverse environments, and a high voltage distribution technique is not applicable to a low voltage home use. Similarly, a technique for distributing source content to web content servers is not germane to a technique for managing or maintaining software products in a client computer system. Thus, a person of ordinary skill in the art would not have been motivated to combine teachings from such a dissimilar operating environment/framework with the teachings of MacInnis.

Even if one were to improperly combine such dissimilar references, there is still no teaching or suggestion of "performing a software maintenance action for the product from a client site by *downloading data* required for said software maintenance action *from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product*, wherein the first subset of files is specific for a given client system". While Saether may teach multiple servers, and the download of data from one server to another, this download of data is merely data *replication* between servers. Such action does not in any way teach or suggest performing a software maintenance action for a product from a client site by downloading data required for said software maintenance from a sequence of repositories *where the sequence of repositories includes at least a top-level repository* *storing a set of files for the product* *and a local-level repository* *storing a first subset of files for the product,* *wherein the first subset of files is specific for a given client system.*

Claim 1 also recites that the data downloaded from the top-level repository is different from data downloaded from the local-level repository. This claimed feature advantageously assists in mitigating network traffic that might otherwise occur in a software maintenance operation (Specification page 17, line 9 – page 18, line 9). In contrast, MacInnis *replicates* data between servers, and thus the data downloaded from a top-level and local-level repository would not be different.

It is therefore urged that the Examiner has failed to properly establish a prima facie showing of obviousness with respect to Claim 1, and such claim has therefore been erroneously rejected under 35 USC 103.

### A.2. Claims 4 and 14

Appellants initially show error in the rejection of Claim 4 for similar reasons to those identified above with respect to Claim 1.

Further with respect to Claim 4, Appellants urge that none of the cited references teach or suggest the claimed feature of "generating an input list of files downloadable *from said sequence of repositories*" (emphasis added). In rejecting this claimed feature, the Examiner states that MacInnis teaches:

> "generating of an input list downloadable from a server repository (e.g. Tabe T, broadcast all versions – col. 4, lines 23-44; Fig. 2)."

Appellants urge that Claim 4 does not merely recite 'a server repository', as alleged by the Examiner in rejecting Claim 4. Rather, Claim 4 explicitly recites 'a *sequence* of repositories' for which a list is generated. To establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03 (emphasis added by Appellants). *See also, In re Royka*, 490 F.2d 580 (C.C.P.A. 1974). If the examiner fails to establish a prima facie case, the rejection is improper and will be overturned. *In re Fine*, 837

F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). As the Examiner has failed to establish a prima facie case of obviousness, as the Examiner has failed to establish, or even allege, a teaching of "generating an input list of files downloadable *from said sequence of repositories*", the rejection is improper. In addition, as the Examiner has failed to establish a prima facie showing of obviousness with respect to Claim 4, the burden has not shifted to Appellants to rebut an obviousness assertion[1].

Still further with respect to Claim 4, Appellants urge that there would have been no reason or other motivation to modify the teachings of the cited references to include the claimed feature of "generating an input list of files downloadable from said *sequence of repositories*", because in both systems (MacInnis and Saether) data is downloaded to a given device from a *single* upstream device so there would be no reason for generating a list of files downloadable from *multiple* sources. This further evidences non-obviousness of Claim 4.

### A.3. Claims 5 and 15

Appellants initially show error in the rejection of Claim 4 for similar reasons to those identified above with respect to Claim 1, and also for reasons given above with respect to Claim 4 (of which Claim 5 depends upon).

Still further with respect to Claim 5, Appellants urge that none of the cited references teach or suggest the claimed feature of "a total input list is generated by *subsequently accessing the repositories and by merging input lists for each repository with a priority of more local files*" (emphasis added by Appellants). In rejecting Claim 5, the Examiner states that MacInnis teaches a version differential matching of input lists (e.g. Fig. 3-5) with a priority of local files (e.g. internal table – col. 4, line 45 to col. 5, line 25), and that Saether discloses "the merge into a delivery list of identified files retrieved from with isolated source servers to yield a final delivery version list for being activated at the target machines (e.g. Fig. 3A)". Appellants urge that none of the above teaching characterizations establishes the claimed 'priority of more local files' as a part of merging input lists. The Examiner states that "the requirement that priority be given to match local files at the target machine is inferred or implicitly disclosed from the teachings of

[1] In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a prima facie case of obviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). Only if that burden is met, does the burden of coming forward with evidence or argument shift to the applicant. Id.

Saether". Appellants urge that this 'inferred or implicit' disclosure is certainly not the case. Saether merely describes a distribution scheme for downloading files *to servers* (secondary global servers and content servers). It does not describe any ability to update *a client computer system* with such files, and thus cannot describe (implicitly or otherwise) a specific operational step that is a part of maintaining software products in client computer systems. Even if the Examiner were to unreasonably interpret Saether's content servers as reading on the claimed client systems, the files distributed to these web content servers are customized based upon the file structure and hardware constraints of the individual content servers (column 1, lines 41-44). Such distribution customization is not in any way based on any priority of local files, but rather is based on the particular hardware configuration of the content server itself. Thus, even with such unreasonable interpretation of Saether's content servers reading on the claimed client systems, there is still no implicit or implied teaching of "*merging input lists for each repository with a* _*priority of more local files*_" as expressly recited in Claim 5. Thus, Claim 5 is further shown to not be obvious in view of the cited references, and has thus been erroneously rejected as a proper prima facie showing of obviousness has not been established.

### A.4. Claims 6, 10 and 16

Appellants initially show error in the rejection of Claim 6 for reasons given above with respect to Claim 1 (of which Claim 6 depends upon).

Further with respect to Claim 6, Appellants urge that none of the cited references teach or suggest the claimed feature of "integrating files into the target system which have been identified by a look-aside procedure as residing in a neighbor system easier to be accessed by the target system than one of said repositories". In rejecting Claim 6, the Examiner states that "Official notice is taken that a search being performed in a network designed so to provide alternative to reach for the nearest node or target point most easily accessible, or to seek out for the least resistive path was a well-known concept in the search algorithm at the time the invention was made". Appellants urge that such "well-known" reasoning in rejecting Claim 6 as being obvious is contrary to well-established law. As stated by the Federal Circuit, "virtually all [inventions] are combinations of old elements." *Environmental Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 698, 218 USPQ 865, 870 (Fed. Cir. 1983); *see also Richdel, Inc. v. Sunspool Corp.*, 714 F.2d 1573, 1579-80, 219 USPQ 8, 12 (Fed. Cir. 1983) ("Most, if not all, inventions are combinations and mostly of old elements."). Therefore an examiner may often find every element of a claimed invention in the prior art. If identification of each claimed element in the prior art were sufficient to negate patentability, very few patents would ever issue. Furthermore, rejecting patents solely by finding prior art corollaries for the claimed elements would permit an examiner to use the claimed invention itself as a blueprint for piecing together elements in the prior art to defeat the patentability of the claimed invention. Such an approach would be "an illogical and inappropriate process by which to determine patentability." *Sensonics, Inc. v. Aerosonic Corp.*, 81 F.3d 1566, 1570, 38 USPQ2d 1551, 1554 (Fed. Cir. 1996). Appellants thus show that the Examiner is using an illogical and inappropriate process in rejecting Claim 6, and thus Claim 6 is shown to have been erroneously rejected based on this "well-known" assertion.

Still further with respect to Claim 6, Appellants urge that the fact that a prior art device could be modified so as to produce the claimed device is not a basis for an obviousness rejection unless the prior art suggested the desirability of such a modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984). There is simply no suggestion of any desire to modify the teachings of the cited references to include the claimed look-aside procedure. The cited MacInnis reference downloads data from a transmitting source 202 to terminals 203 and 204 in a unidirectional fashion (Figure 2; column 4, lines 22-44), such that the terminals do not have to request content from the source (column 4, lines 36-44). Because of the uni-directional transmission of content, there is no ability for the terminal to somehow determine that a neighbor system can be more easily accessed than one of the repositories. Quite simply, cable TV networks do not have any ability to download data or content from one neighboring terminal to another. Thus, a person of ordinary skill in the art, when presented with the teachings of MacInnis, would not have been motivated to modify the teachings therein in accordance with the claimed look-aside procedure.

As to the teachings of Saether, such reference merely teaches content distribution to content *servers*, and does not describe any technique for maintaining software products in *client* computer systems, and expressly states a desire to update such content servers from an upstream repository, as the upstream repository customizes/tailors the distribution of the set of source files according to the specific content server configuration (column 1, lines 41-44)). Thus, there is no suggestion of any desire to modify the teachings of this cited Saether reference to include the claimed look-aside procedure. In fact, modifying the teachings of Saether in accordance with the claimed invention would eviscerate the expressed purpose of such teachings, as the file distribution would not be customized per the unique configuration of the content server. This strongly evidences no motivation to modify such teachings in accordance with the claimed invention recited in Claim 6. Accordingly, per the holding in *In re Gordon, Id.*, Claim 6 is further shown to have been erroneously rejected.

In conclusion, Appellants request that the Board reverse the rejection of Claims 1-16, as the Examiner has failed to properly establish a prima facie showing of obviousness with respect to such claims.

Duke W. Yee
Reg. No. 34,285
Wayne P. Bailey
Reg. No. 34,289
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

## CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A method for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution wherein the client computer systems are connectable with the at least one central software maintenance institution via a network, the method comprising the steps of:

providing product information for a product in the network system for making the product information available for said plurality of client systems; and

performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system, and data downloaded from the top-level repository is different from data downloaded from the local-level repository and the data downloaded from both the top-level repository and the local-level repository is used by the given client system in performing the software maintenance action.

2. The method according to claim 1 wherein the sequence of repositories includes a mid-level repository storing a second subset of files for the product, wherein the second subset of files includes at least one of a version update, a fix, and nation-specific files.

3. The method according to claim 2 in which a fall back to an older program version is achieved by inactivating a newer version and activating the older version

4. A method for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution wherein the client computer systems are connectable with the at least one central software maintenance institution via a network, the method comprising the steps of:

providing product information for a product in the network system for making the product information available for said plurality of client systems;

performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system, wherein the performing step comprises:

generating an input list of files downloadable from said sequence of repositories;

generating a list of files present on said target client system;

comparing the list of files downloadable from said sequence of repositories with the list of files present on said target client system; and

downloading a plurality of files, wherein the plurality of files includes only files which are not yet present in the target client system.

5. The method according to claim 4 in which a total input list is generated by subsequently accessing the repositories and by merging input lists for each repository with a priority of more local files.

6. The method according to claim 1 further comprising the step of integrating files into the target system which have been identified by a look-aside procedure as residing in a neighbor system easier to be accessed by the target system than one of said repositories.

7. A system for maintaining software products, the system comprising:

at least one central software maintenance site;

a network;

a plurality of client computer systems decentralized relative to the at least one central software maintenance site, wherein the client computer systems are connectable with the at least one central software maintenance institution via the network; and

a sequence of repositories, wherein the sequence of repositories provides product information for a product in the network system for making the product information available for said plurality of client systems, wherein said sequence of repositories includes at least a top-level repository storing a complete set of files for the product and a local-level repository storing a first subset of files for the product, wherein the subset of files is specific for a given client system,

wherein a given client computer system from within the plurality of client computer systems performs a software maintenance action for the product by downloading data required for said software maintenance action from the sequence of repositories and data downloaded from the top-level repository is different from data downloaded from the local-level repository and the data downloaded from both the top-level repository and the local-level repository is used by the given client system in performing the software maintenance action.

8. The system according to claim 7, wherein the sequence of repositories is provided as a plurality of hierarchically arranged repositories.

9. The system according to claim 7, wherein the sequence of repositories includes a mid-level repository storing a second subset of files for the product, wherein the second subset of files includes at least one of a version update, a fix, and nation-specific files.

10. The system according to claim 8, further comprising:

at least one neighbor system, wherein the software maintenance action includes integrating files into the target system which have been identified by a look-aside procedure as residing in the at least one neighbor system easier to be accessed by the target system than one of said repositories.

11. The system according to claim 7, further comprising shadow repositories for at least a subset of the sequence of repositories.

12. A computer program product, in a computer readable medium, for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution wherein the client computer systems are connectable with the at least one central software maintenance institution via a

network, the computer program product comprising:

instructions for providing product information for a product in the network system for making the product information available for said plurality of client systems; and

instructions for performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a complete set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system, and data downloaded from the top-level repository is different from data downloaded from the local-level repository and the data downloaded from both the top-level repository and the local-level repository is used by the given client system in performing the software maintenance action.

13. The computer program product according to claim 12, wherein the sequence of repositories includes a mid-level repository storing a second subset of files for the product, wherein the second subset of files includes at least one of a version update, a fix, and nation-specific files.

14. The computer program product according to claim 13 in which the instructions for performing said maintenance action serves for an upgrade of a program on at least one target system and the instructions for performing said maintenance action includes:

instructions for generating an input list of files downloadable from said sequence of repositories;

instructions for generating a list of files present on said target client system;

instructions for comparing the list of files downloadable from said sequence of repositories with the list of files present on said target client system; and

instructions for downloading a plurality of files, wherein the plurality of files includes only files which are not yet present in the target client system.

15. The computer program product according to claim 14 in which a total input list is generated by subsequently accessing the repositories and by merging input lists for each repository with a priority of more local files.

16. The computer program product according to claim 12, further comprising instructions for integrating files into the target system which have been identified by a look-aside procedure as residing in a neighbor system easier to be accessed by the target system than one of said repositories.

# EVIDENCE APPENDIX

There is no evidence to be presented.

## RELATED PROCEEDINGS APPENDIX

There are no related proceedings.

**Yee &
Associates, P.C.**

4100 Alpha Road
Suite 1100
Dallas, Texas 75244

Main No. (972) 385-8777
Facsimile (972) 385-7766

# Facsimile Cover Sheet

| | |
|---|---|
| To: Commissioner for Patents for<br>**Examiner Tuan A. Vu**<br>**Group Art Unit 2193** | Facsimile No.: **571/273-8300** |
| From: Jennifer Pilcher<br>Legal Assistant to Wayne Bailey | No. of Pages Including Cover Sheet: 25 |

Message:

Enclosed herewith:

- Transmittal Document; and
- Appeal Brief.

Re: Application No. **09/696,399**
Attorney Docket No: **DE9-1999-0075US1**

Date: Monday, August 22, 2005

**Please contact us at (972) 385-8777 if you do not receive all pages indicated above or experience any difficulty in receiving this facsimile.**

*This Facsimile is intended only for the use of the addressee and, if the addressee is a client or their agent, contains privileged and confidential information. If you are not the intended recipient of this facsimile, you have received this facsimile inadvertently and in error. Any review, dissemination, distribution, or copying is strictly prohibited. If you received this facsimile in error, please notify us by telephone and return the facsimile to us immediately.*

**PLEASE CONFIRM RECEIPT OF THIS TRANSMISSION BY FAXING A CONFIRMATION TO 972-385-7766.**